

TECNOLOGIAS DE DESENVOLVIMENTO CROSS-PLATFORM

GONÇALO CAMPOS PEREIRA, PG53834

SANTIAGO DOMINGUES, PG54225

UNIVERSO MOBILE

A utilização de dispositivos mobile tem aumentado exponencialmente ao longo dos últimos anos

- Expectativa (de performance e esforço)
- Influência e utilidade sociais
- Entretenimento
- Comunicação

"Worldwide, smartphones were used by 1.85 billion people in 2014. This number is expected to be 2.32 billion in 2017 and 2.87 billion in 2020"

H. Brito, A. Gomes, Á. Santos and J. Bernardino, "JavaScript in mobile applications: React native vs ionic vs NativeScript vs native development," 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), Caceres, Spain, 2018, pp. 1-6, doi: 10.23919/CISTI.2018.8399283.

DESENVOLVIMENTO MOBILE

- Complexo:
 - Grande variedade de marcas e dispositivos (**Apple, Samsung, Xiaomi**)
 - Múltiplos e diferentes sistemas operativos (**Android, iOS, macOS, Windows**)
 - Gestão de recursos limitados, como memória e bateria, em dispositivos móveis
 - Necessidade de adaptação e otimização específica para cada sistema operativo e dispositivo

ABORDAGENS AO DESENVOLVIMENTO MOBILE:

- Necessidade de criar aplicações separadamente para cada plataforma, como iOS e Android
- Requerem diferentes abordagens no seu desenvolvimento

Problemas?

- extremamente custosa em termos da manutenção, recursos e tempo.
- pode causar inconsistências entre as diferentes versões desenvolvidas

ABORDAGENS AO DESENVOLVIMENTO MOBILE:

Para combater este desafio surgiram várias frameworks:

- React Native
- Flutter
- Xamarin
- Kotlin Multiplatform
- **NATIVESCRIPT**

Diferentes vantagens e desvantagens!

permite aos desenvolvedores conceber aplicações
para as diferentes plataformas/sistemas operativos
a partir de num único código-fonte

NATIVESCRIPT

Framework open-source para desenvolver aplicações móveis nas plataformas iOS e Android.

1. Lançada em 2014
2. Desenvolvida pela *Progress*

NATIVESCRIPT - LINGUAGENS

- *Em dezembro de 2020 foi feita a integração do NativeScript na OpenJS Foundation*
 - Aplicações construídos em **JavaScript**
 - Qualquer linguagem que seja compilada e traduzida para **JavaScript**
 - o **TypeScript**

Oferece suporte às frameworks **JavaScript Angular** e **Vue**

NATIVESCRIPT – CONHECIMENTO PRÉVIO

- Linguagens de programação (**Javascript** e **Typescript**)
- Desenvolvimento Web e Mobile (**XML**, **CSS**)
- Plataformas de desenvolvimento mobile (**Xcode**, **Android Studio**)
- APIs Nativas
- Arquitetura, Plugins e ferramentas de desenvolvimento

NATIVESCRIPT – SDA (SOFTWARE DEVELOPMENT APPROACH)

De notar
que existe também
uma porção
de desenvolvimento
Web (com recurso a
XML e CSS)

- Abordagem nativa:
 - o Procura que as aplicações criadas com recurso à mesma sejam compiladas em código específico para cada plataforma
 - o Permite aos desenvolvedores criar aplicações que têm acesso total às APIs nativas do dispositivo
 - o Compilação de código JavaScript/TypeScript para código nativo para cada plataforma alvo (iOS e Android)
 - o Permite ao utilizador uma experiência completamente nativa

NATIVESCRIPT – DIAGRAMAS ESTRUTURAIS

- Código do aplicação em Javascript
- Em vez de recriar componentes da interface com HTML, a framework instancia elementos de visualização nativos.
- Estes são definidos de forma declarativa usando XML
- O estilo e as animações são aplicados usando CSS.

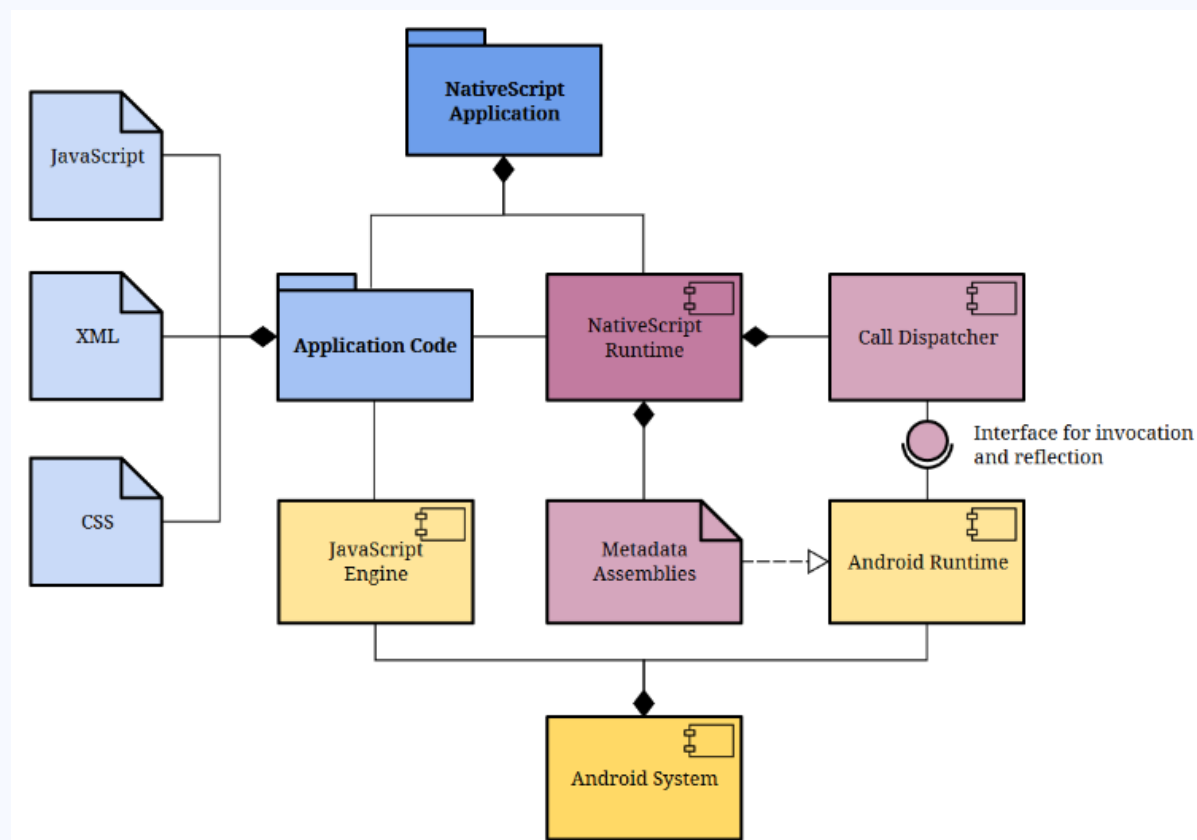


Diagrama de componentes UML - arquitetura NativeScript e integração com o sistema Android

NATIVESCRIPT – DIAGRAMAS ESTRUTURAIS

- O código da aplicação é executado dentro do engine JavaScript do sistema.
- Em preparação para a sua execução, são injetados proxies para as interfaces do sistema no namespace do JavaScript.
- Com esse engine, a aplicação é executada de forma muito próxima de como uma aplicação nativa executaria.

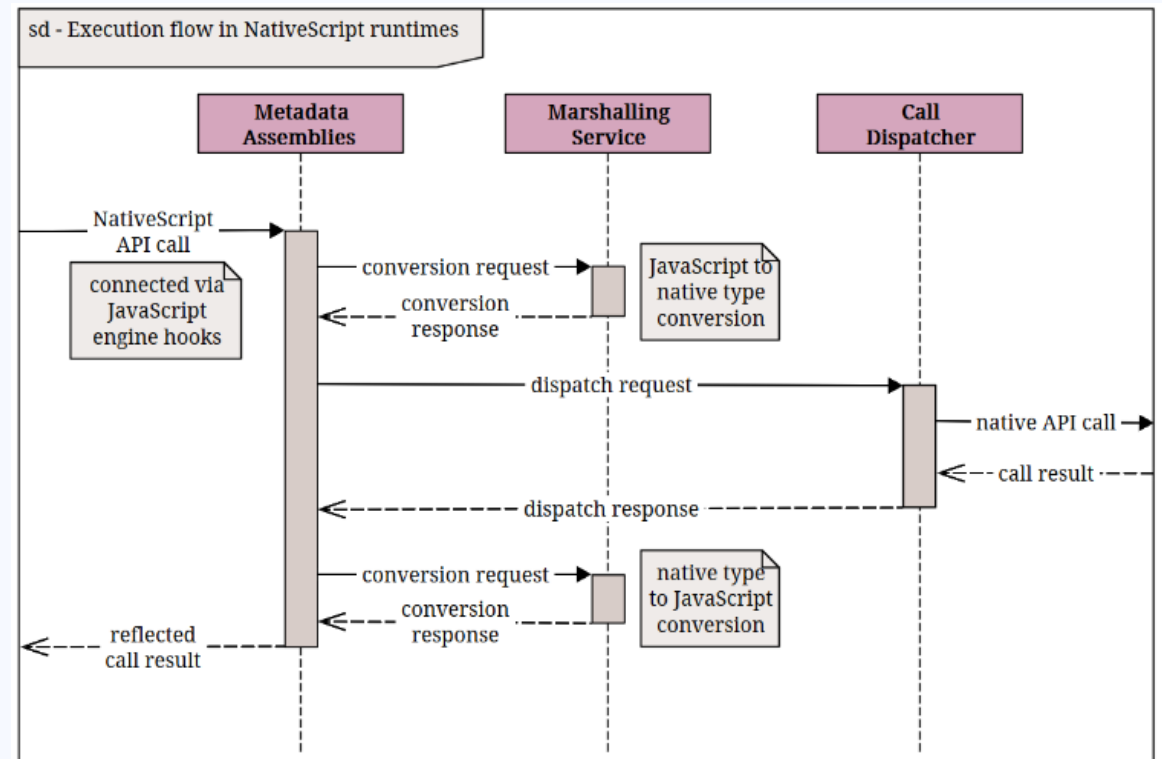


Diagrama de sequência UML - Fluxo de execução em runtime NativeScript

NATIVESCRIPT – VANTAGENS E LIMITAÇÕES

VANTAGENS:

- Partilha de código entre iOS e Android
- As aplicações são compiladas em código nativo, proporcionando desempenho e experiência de utilizador semelhantes a aplicações nativas
- Os desenvolvedores têm acesso total às APIs nativas de cada plataforma, permitindo otimizar a aplicação, ou seja, permite uma boa integração com o sistema operativo e com o hardware
- Suporte nativo para TypeScript, proporcionando aos desenvolvedores uma experiência de desenvolvimento mais robusta

NATIVESCRIPT – VANTAGENS E LIMITAÇÕES

LIMITAÇÕES:

- As aplicações NativeScript podem ser um pouco maiores do que aplicações puramente nativas, devido à necessidade de incluir o engine do JavaScript
- A quantidade de documentação é relativamente baixa
- Pode ser necessário depender de plugins ou escrever código nativo, e o acesso às APIs em alguns casos específicos pode ser limitado.

NATIVESCRIPT - USE CASES E APLICABILIDADE

- **Aplicações de consumo** : aplicações que se destinam a utilizadores finais e não empresas.
 - Redes Sociais - aplicações com interações responsivas e animações fluídas
- **Aplicações de entretenimento:**
 - Serviços de reprodução e/ou streaming - pretende-se que a navegação e a descoberta de conteúdo seja otimizada para cada sistema operativo
- **Aplicações projetadas para auxiliar a organização e realização das tarefas do quotidiano dos utilizadores**
 - o Neste caso, o acesso às APIs pode ser útil quer para guardar informações na nuvem do utilizador, como para gerar notificações

NATIVESCRIPT - USE CASES E APLICABILIDADE

- **Aplicações de Serviços:** ideal para o desenvolvimento de aplicações de serviços, como aplicações de entrega de comida, de reservas de hotéis, de transporte
 - o oferece a capacidade de integrar recursos nativos, como geolocalização, notificações push, pagamentos móveis entre outros.
- **Aplicações Empresariais Internas:** adequado para o desenvolvimento de aplicações corporativas internas, como aplicações de gestão de stock, rastreamento de ativos, aplicações de Recursos Humanos e aplicações de comunicação interna

NATIVESCRIPT - SISTEMAS/APLICAÇÕES

- **JScrambler** - ferramenta de segurança para aplicações da web, especialmente para JavaScript. Oferece recursos para proteger o código-fonte JavaScript, dificultando atividades como a engenharia reversa e a cópia não autorizada. Fornece também técnicas avançadas de proteção contra ataques de injeção de código e adulteração de código JavaScript
- **PayPal Commerce Platform** - Solução oferecida pela PayPal para ajudar empresas a aceitar pagamentos online e a gerir suas operações de comércio eletrónico.
- **Strudel** - Aplicação para gestão de tarefas e lista de compras.
- **Deliveroo** - Aplicação de entrega de comida que opera em vários países.
- **Le cab** - Aplicação de reserva de táxis (França).

NATIVESCRIPT - COMPARAÇÃO COM OUTRAS ABORDAGENS

NATIVESCRIPT VS REACT NATIVE

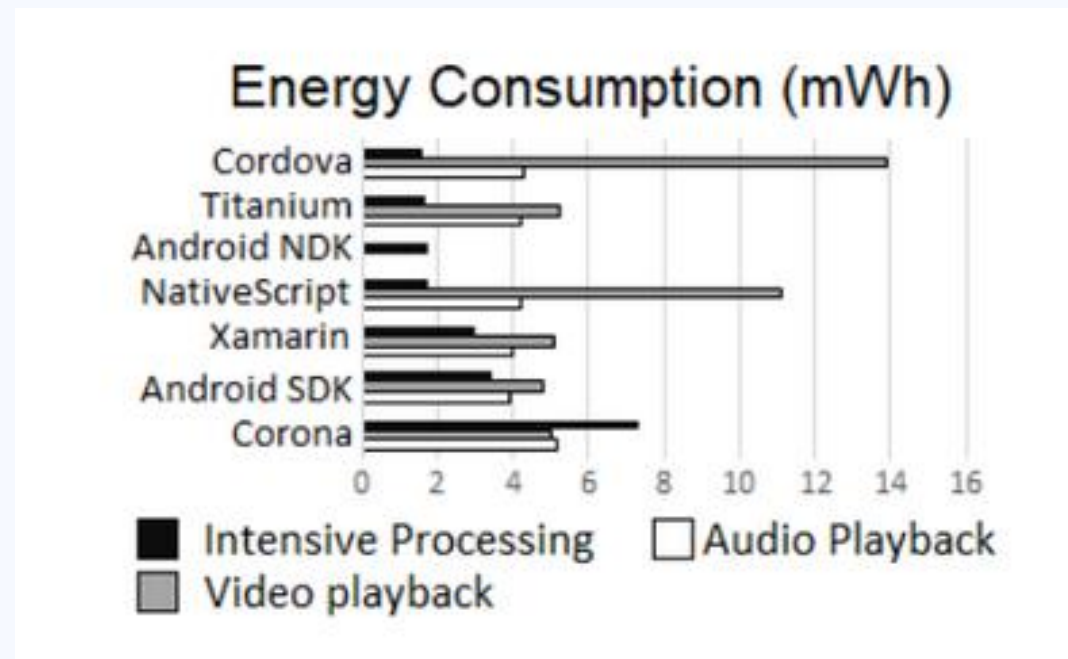
Consumption type	NativeScript	React Native	Difference
Total	39.96	19.96	95.93%
Allocated	24.10	18.57	29.84%

APK NATIVESCRIPT – 16.4 MB
APK REACT NATIVE – 7.87 MB

Build type	NativeScript	React Native	Difference
Full compilation	02:17,03	01:19,77	71.77%
Successive compilation	00:21,27	00:24,15	–11.93%
Livesync view	00:04,90	00:00,96	410.21%
Livesync code	00:11,58	00:01,02	1031.05%

Veselý, B. D. Analysis and experiments
with NativeScript and React Native
framework

NATIVESCRIPT - COMPARAÇÃO COM OUTRAS ABORDAGENS



Corbalán, Leonardo & Fernandez Sosa, Juan & Cuitiño, Alfonso & Delia, Lisandro & Cáseres, Germán & Thomas, Pablo & Pesado, Patricia. (2018). Development frameworks for mobile devices: a comparative study about energy consumption. 191-201. 10.1145/3197231.3197242.

NATIVESCRIPT - COMPARAÇÃO COM OUTRAS ABORDAGENS

Critérios	Soluções			
	<i>Nativas</i>	<i>React Native</i>	<i>Native Script</i>	<i>Ionic</i>
Aprendizagem e qualidade da documentação	2	5	3	4
Custo de Desenvolvimento	2	4	4	4
Emuladores e Depuração	4	5	4	4
Tempo de resposta e velocidade	5	5	4	1
Reconhecimento comercial	5	5	1	2
Reutilização de código e trabalho em equipa	2	5	5	4
Manutenção e Atualizações	2	3	4	4
TOTAL	22	32	25	23

ANÁLISE DOS CRITÉRIOS NA ESCALA DE LIKERT

(Os critérios seleccionados foram escolhidos de acordo com as necessidades principais do desenvolvimento móvel)

H. Brito, A. Gomes, Á. Santos and J. Bernardino, "JavaScript in mobile applications: React native vs ionic vs NativeScript vs native development," 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), Cáceres, Spain, 2018, pp. 1-6, doi: 10.23919/CISTI.2018.8399283.

TECNOLOGIAS DE DESENVOLVIMENTO CROSS-PLATFORM

GONÇALO CAMPOS PEREIRA, PG53834

SANTIAGO DOMINGUES, PG54225